System Security
Constructor University
Dr. Jürgen Schönwälder

Module: MCSSE-CYB-02
Date: 2026-03-20
Due: 2026-03-27

Problem Sheet #5

Problem 5.1: *security enhanced linux* (2+2+2+2+2 = 10 points)

Security Enhanced Linux (SE Linux) provides mandatory access control for Linux systems. SE Linux allows security administrators to define security policies that are loaded into the kernel and then enforced. SE Linux policies confine user programs and system services, as well as access to files and network resources.

James Freeman has published a sequence of hands-on lab introducing SE Linux. You are asked to work through the labs and to answer the following questions.

Note: If you do the labs on a Debian or Ubuntu system, you will encounter some minor differences and not all tools may be available as packages.

a) Briefly describe what the test program `testprog` is doing.

b) What is the purpose of the files `testprog.fc` and `testprog.fs`? How do these files relate to `testprog.pp`? Which programs are invoked by the Makefile to generate `testprog.pp`?

c) After setting the proper file context, the program fails with a segmentation fault. Run strace on the command. Which system call fails? Can you confim this in the audit log?

d) What is the security label of the systemd init process (pid 1)? What is the security label of testprog after launching it via systemd? Indicate the operating system and version you are using.

e) What is the purpose of the `testprog.if` file? How are the definitions provided by the `testprog.if` file used by `testcat.te`?

Special notes on Debian:

- For `testprog`, I had to add the following to the `testprog.te` file:

```
require {
      # ...
      class fd { use };
      class chr_file { read write };
}

allow testprog_t testprog_exec_t:file map;
allow testprog_t unconfined_t:fd use;
allow testprog_t user_devpts_t:chr_file { read write };
```

- For `testcat`, I had to add the following to the `testcat.te` file:

```
require {
      # ...
      type user_devpts_t;
      class fd { use };
      class chr_file { read write };
}

allow testcat_t testcat_exec_t:file map;
allow testcat_t unconfined_t:fd use;
allow testcat_t user_devpts_t:chr_file { read write };
```